Planning and Scheduling for Fleets of Earth Observing Satellites

Jeremy Frank, Ari Jónsson*, Robert Morris, David E. Smith
Computational Sciences Division
NASA Ames Research Center, MS 269-2
{frank,jonsson,morris,de2smith}@ptolemy.arc.nasa.gov
Moffett Field, CA 94035 †

Abstract—
We address the problem of scheduling observations for a collection of Earth Observing Satellites. This scheduling task is a difficult optimization problem, involving multiple satellites, hundreds of requests, constraints on when and how to service each request, and resources such as instruments, recording devices, transmitters, and ground stations. High-fidelity models are required to ensure the validity of schedules. At the same time, the size and complexity of the problem makes it unlikely that systematic search methods will be able to produce good solutions in a reasonable amount of time. This paper presents an approach to solving the Earth Observing Satellite scheduling problem that involves: 1) modeling of the problem using a constraint-based language, 2) a stochastic greedy search algorithm for finding solutions, and 3) heuristics based on a generalized contention measure for guiding the search.

Keywords: Planning, scheduling, stochastic search, constraint satisfaction, Earth observing satellites.

I. Introduction

NASA's growing fleet of Earth Observing Satellites (EOSs) [1] employ advanced sensing technology to assist scientists in the fields of meteorology, oceanography, biology, geology, and atmospheric science to better understand the complex interactions among Earth's lands, oceans, and atmosphere. Demand on these satellites is already high, and is expected to increase significantly in the near future. Currently, science activities on different satellites (e.g. the AM Constellation) or even different instruments on the same satellite (e.g. the ASTER instrument on the Terra satellite [11]), are scheduled independently of one another, requiring the manual coordination of observations by communicating teams of mission planners.

It is unlikely that this approach to daily mission planning and scheduling will be viable in the future. As the number of satellites and the number of observation requests grow large, manual coordination will no longer be possible. A more effective way to manage observation scheduling is by allowing customers of the data (viz. the scientists themselves) to request data products from a central authority instead of an individual satellite or mission. Customer preferences will constrain which satellite or satellites will be used to collect the data. Automated techniques can reason about all of the resources that are involved in collecting data, storing the data temporarily on board satellites, and transmitting the data back to Earth. This will enable more efficient management of the fleet of satellites as well as the communication resources that support them.

In this paper we discuss the problem of scheduling observations for a collection of Earth Observing Satellites. In Section 2 we formulate the problem as a constrained optimization problem, involving a set of observation requests, each with associated constraints that must be satisfied by any solution to the problem, and a set of resources, including imaging instruments, solid state recorders (SSRs), antennae, transmitters, and ground stations. Typically, there will be too many observations to schedule with available satellite resources. Therefore, we assume requests are prioritized, and search for the *best* subset of requests to service, subject to operational constraints. In Section 3, we survey approaches to solving the EOS scheduling problem. In Section 4, we introduce our underlying Constraint-Based Interval (CBI) representation [16]. In CBI, actions and fluents (or states) are uniformly described as intervals during which a state variable maintains a particular value. The CBI representation allows modeling of how actions and fluents are related to each other in a plan. Candidate plans are represented by variables and constraints which reflect the temporal relationships between actions, ordering decisions between actions, and constraints on the parameters of states or actions. In Section 5 we describe our approach to searching for observation schedules. In particular, we use a stochastic greedy search algorithm based on the Heuristic Biased Stochastic Sampling (HBSS) algorithm [3]. In Section 6 we describe the heuristics used to guide this search procedure. They are based on a generalized contention measure that helps to estimate the difficulty of scheduling individual observations. In Section 7 we con-

*Research Institute for Advanced Computer Science (RIACS)

†The authors appreciate the helpful contributions to this work made by Jennifer Dungan. This work was funded by NASA's Advanced Information Systems Technology Program, under contract AIST-0042.

[1] We will use EOS to refer to general Earth Observing Satellites, not to be confused with the specific EOS satellite.

clude and discuss future work.

II. Problem Description

We assume that constellations of the future will contain many satellites with heterogeneous capabilities. The satellites may be in any orbit. Each satellite is equipped with a suite of instruments. Some satellites will have pointable instruments, giving increased flexibility in what they can observe at any point in an orbit. Some instruments, like synthetic aperture radar, can be used in all visibility conditions, while others can only be used on lighted regions of the Earth. Other instruments may have overlapping spectral capabilities. Satellites will also have varying SSR capacity.

Image data acquired by an Earth Observing Satellite are either downlinked in real-time, or recorded on board for playback at a later time. Ground stations and TDRSS satellites are available to receive downlinked images. Different satellites may be able to communicate with only a subset of these resources, and transmission rates will differ from satellite to satellite and from station to station. Further, there may be different financial costs associated with using different communications resources.

An observation request is typically specified in terms of the type of data and instrument desired, and a series of locations and times for the sensing event. A priority, corresponding to the scientific utility of the data, is also assigned to the request. A proposed observation sequence must satisfy a number of constraints. These constraints include requirements on the instruments used to collect the data, and duration and ordering constraints associated with the data collecting, recording, and downlinking tasks. In addition, SSR capacity, and constraints on communications equipment such as satellite antennae and ground stations must be satisfied. There may also be set-up steps associated with particular operations, like establishing a data link prior to downlink, or aiming an instrument prior to data acquisition. These steps generate further temporal and ordering constraints. A request can also involve coordinating activities among different satellites. For example, a stereo image will involve multiple sensing events of the same location at different viewing angles. In other cases, adequate spectral coverage may require the use of two or more instruments to sense the same land area, or to sense both land use and atmospheric conditions. Finally, scientists may want to image the same area at different times of day.

Often there will be too many observations to schedule with available satellite resources. Solutions are preferred based on objectives such as maximizing the number of high priority requests serviced, maximizing the expected quality of the observations, and minimizing the cost of downlink operations.

In the EOS scheduling domain, requests can be submitted at any time, and high priority targets of opportunity (e.g., fires, earthquakes, volcanos) may result in the need for revising a partially executed schedule. In addition, there are numerous sources of uncertainty. One of the most important, and difficult, aspects of the EOS scheduling problem arises from the uncertainty of the weather, specifically, with respect to cloud cover. Image quality is determined by the amount of cloud cover and many parts of the world have long seasons where clouds are omnipresent. If a simple "no cloud" scheduling policy were followed, these parts of the world would virtually never be observed. Thus, it is important to enforce a sophisticated scheduling policy which balances a "no cloud" cover restriction with the need for coverage.

III. Previous Work

Previously reported work on EOS scheduling problems includes both theoretical investigations using abstract models, as well as operational schedulers for ongoing EOS missions. We divide our survey of previous approaches into two parts: modeling and algorithms.

A. Models of EOS Scheduling

Very few approaches consider multiple satellites or the coordination of observations. Burrowbridge [4] discusses the problem of managing telemetry and data acquisition (TDA) resources needed by multiple satellites, but does not treat problems involving observations, data gathering, or downlinking data.

There are a number of theoretical studies on managing a single satellite. These usually involve simplified models of the satellite. For example, Lemâitre et al. [10], Pembert on [12] and Wolfe and Sorensen [18] do not discuss on-board data storage or communications system management. Bensana et al. [2] describe problems with on-board storage constraints, but without communications system management. Pemberton [12] and Wolfe and Sorensen [18] assume that there are no precedence constraints or any other logical constraints between the requests, while Lemâitre et al. [10] and Bensana et al. [2] compile the complex constraints down to simple binary and trinary exclusion constraints.

There are several operational systems for ongoing EOS missions. The ASTER scheduler described in [11] and the Landsat 7 scheduler [13] are two examples. These schedulers have quite detailed models of the satellites and the communications environment. However, they do suffer from some limitations. For example, ASTER scheduling is performed independently of other instruments onboard the Terra satellite. A fixed amount of memory is allocated for this instrument; if it is unused, it can't be used by any other instrument, resulting in suboptimal schedules. Additionally, these models do not account for all of the steps that occur on board the satellites. For instance, the ASTER instrument is aimable, yet there is no accounting for the time required to aim the instrument between observations. Similarly, Landsat requires time to shut down and power up its instrument; this is assumed to take place between scenes. While this may be sufficient for Landsat, it may not be good enough for future satellites with more advanced capabilities. The ASPEN planning system has been used to schedule observations for EO-1 [14, 15]. However, the scheduling problem described in [15] does not appear very difficult; EO-1 can only schedule 4 observations a day. It is not clear how their approach scales to many satellites with many instruments of varying capabilities.

As mentioned previously, most of the problems described in these papers are optimization problems. The usual goal is to maximize the weighted sum of the scheduled observations. Wolfe and Sorensen [18] describe a slightly different problem, in which observations are valued based on when they are performed and how much data is collected.

B. Scheduling Algorithms

Many of the search algorithms described in the literature are incomplete algorithms. The primary reason for focusing on such algorithms is that, even for small numbers of satellites, the problems are large enough that solving them optimally is impractical. The usual approach is to greedily select the next highest priority request to try and schedule, and reject it if there is nowhere for it to go. The ASTER scheduler [11] works exactly this way, as does an approach described by Wolfe and Sorensen [18]. Pemberton [12] describes a family of algorithms ranging from strictly greedy to complete search; after sorting the requests, blocks of n requests are scheduled optimally, with all previous allocations acting as constraints on the next set of observations to schedule. Burrowbridge [4] greedily schedules requests based on the earliest finishing time of the request. The Landsat 7 scheduler [13] greedily schedules requests based on the earliest finishing time until resources run out, then preempts previously scheduled observations based on priority. ASPEN [14] uses a local search algorithm that generates an initial schedule, then identifies and repairs conflicts in the schedule by changing variable assignments. This algorithm is quite complex, with 10 distinguished types of conflicts, and many heuristics required to identify both the conflict to work on and the method of addressing it.

IV. Representation

We believe that effective coordination of EOSs requires high-fidelity modeling of the entire EOS environment. Not only do we need to model on-board satellite resources, communication resources and requests, but we must also model the detailed activity sequences on the spacecraft and on the ground. However, we would like to make use of search techniques developed for solving combinatorial problems. To balance these needs, we use the Constraint-Based Interval (CBI) framework.

A. Constraint-Based Interval Representation

The CBI framework [16] is based on an interval representation of time. A *predicate* is a uniform representation of actions and states, and an *interval* is the period during which a predicate holds. A *token* is used to represent a predicate which holds during an interval. Each token is defined by the start, end and duration of the interval where it occurs, as well as other parameters which further elaborate on the predicate. For instance, a `Take-Image` predicate has a parameter `?mode` describing the gain, which can be either `low` or `high`. The relationships between different activities are described by *planning schemata* which specify, for each token, other tokens that must exist (e.g. pre and post conditions), and how the tokens are related to each other.

```
(Define_Compatibility
(SINGLE ((Instrument Camera))
    ((Take-Image(?angle, ?mode, ?init, ?final))))
:parameter_functions
    ((Data-Level(?init, ?final, ?mode, ?_duration_)))
:compatibility_spec
    (AND
      (meets
      (SINGLE ((Instrument Camera))
        (Idle)))
      (?mode OR
        (HIGH met_by
          (SINGLE ((Instrument Camera))
            ((Calibrate(?angle))))))))
```

Figure 1: DDL syntax for a planning schema (called a *Compatibility* in DDL). The master activity for this schema is a `Take-Image`. The :parameter_functions portion of the schema lists constraints on the parameters of the activity. The :compatibility_spec describes other activities which must exist when the master activity is in the plan. A conditional activity is shown using the OR construction; the `Calibrate` activity depends on the value of the ?mode parameter.

We use the Domain Description Language (DDL) [9] to specify planning schemata. Figure 1 shows an example of a planning schema written in DDL. Schemata can specify conditional effects and disjunctions of required tokens. For instance in Figure 1, a `Take-Image` interval can be met by a `Calibration` period if a high resolution image is to be taken. The value of the `?mode` parameter indicates whether or not a `Calibration` period is required. Planning schemata can also include constraints on the parameters of the token. As shown in Figure 1, the `Take-Image` interval has a constraint relating the mode and the amount of data stored by the operation.

EUROPA [6] is a CBI planning paradigm which continuously reformulates the planning problem as a Dynamic Constraint Satisfaction Problem (DCSP). This is done by mapping each partial plan to a CSP. The temporal constraints form a Simple Temporal Network, which can be efficiently solved [5], while the rest of the constraints form a general, non-binary CSP represented by procedural constraints [8]. An additional feature includes the ability to produce plans with flexible time; that is, activities may start and end at any time in an interval [9]. This gives the plan some flexibility, should activities take longer or shorter than expected. Figure 2 shows a plan fragment and its induced CSP. Assignments of variables in the CSP correspond either to the adding of new plan steps, or the assignment of parameters of plan steps. As steps are added to or removed from the plan, the CSP is updated to reflect the current partial plan. For example, in Figure 2, adding the `Take-Image` step to the plan requires adding several new variables and constraints to the CSP. At any time, if the CSP is inconsistent, then the partial plan it represents is invalid; if a solution is found to the CSP, then that solution can be mapped back to a plan which solves the problem. The advantage of such a representation is that any algorithm which solves DCSPs can be used to solve the planning problem.

Figure 2: A partial plan and its DCSP representation. The partial plan consists of 2 activities, shown at the top of the figure. The DCSP variables are in rounded boxes. Edges between DCSP variables are labeled with the constraints on those variables.

EUROPA has the ability to model various types of resources. A domain model consists of a number of *attributes*, each of which represents an aspect of the objects that interact in the world. Each of these attributes may be in only one state at a time; hence, if a camera is taking an image, it can't also be turning. This permits simple modeling of resources. Complex resources such as fuel and power can be modeled using numerical constraints. In Figure 1, the filling of the SSR is modeled by a constraint that relates the initial amount of storage, the final amount of storage, and the rate at which the data acquisition task fills the buffer.

B. A CBI Model of the EOS Domain

A CBI model for the EOS domain will describe the attributes of a set of satellites with different types of sensing instruments and resources, as well as different orbital tracks. Resources to be modeled for each satellite include the instruments, the SSR, and a set of antennae and transmitters for downlinking data. We do not explicitly model power consumption or satellite maneuvering operations, although maneuvering periods and power-related duty cycles may constrain the schedule. Other model elements are data receiving stations, either ground stations or TDRSS satellites.

A sensing instrument is defined primarily in terms of the type of data it acquires, the spatial and spectral resolution of eveery waveband (for imaging spectrometers), its swath width, and pointing limitations (field of view, slew rate, and so on). A solid state recording device (SSR) is defined by the storage capacity and the rate at which it stores data. Antennae and transmitting devices are defined by whether they are slewable, and also by their data transmission rate. Data receiving stations are associated with a frequency band, and also by the number of downlink channels they support. Each of these entities will correspond to one or more attributes of a model.

Figure 3: A simplified model showing the interaction of instrument and SSR attributes. Ovals represent the states permitted for each attribute. Solid lines indicate possible state transitions within an attribute, dashed lines indicate temporal constraints required between attributes, and boxes indicate constraints on the parameters of certain state.

Requests are identified by their location, either specified in World Reference System (WRS) units, or latitude and longitude. We can also model a "Quality of Service" (QoS) type for each request. This would allow customers to pay less for data of either lower quality or longer delivery time. For example, in Landsat 7, requests for images made by non-U.S. international ground stations are usually serviced through direct downlink to the the requesting ground station. By contrast, so-called "special" requests on Landsat 7 corresponding to exceptional events are typically simultaneously recorded and directly downlinked to a ground station, and later also played back for redundancy. The utility of scheduling a request at a particular time is a function of both the user-defined priority and the conditions; for example, clouds may decrease the quality of data for an observation. Thus, there is a constraint relating the conditions, priority, and the utility of performing the observation under those conditions. A given request may also correspond to a coordinated activity involving multiple instruments. Coordinated observation activities arise for many reasons, for example, to take a stereo image of an area, to sample a region over different spectral regions, or to calibrate instruments.

Figure 4: A complete EUROPA plan with two EOSs and one TDRSS. Each satellite has 3 attributes: the instrument, SSR, and communication antenna. The TDRSS has two attributes: the contact and transmitter.

Each attribute of a CBI model supports a limited set of activities. Thus, an SSR can be recording, playing back data, or idle; an antenna can be slewing, or pointing to a receiving station; and an imaging instrument can be off, idle, or taking an image. The model will also represent set up events such as warming up an instrument, or slewing for antennae or pointable sensing instruments. Temporal constraints impose restrictions on the duration and ordering of tokens in a plan. Temporal constraints may be associated with a single activity, such as the constraint that an antenna be slewed to a certain location before it can be pointing at that location; or a temporal constraint can involve pairs of activities, such as the constraint that a ground station must be in contact with a satellite while data is being downlinked. Resource constraints include SSR capacity, communication bandwidth, and duty cycle restrictions on imaging instruments. Figure 3 shows how all of these aspects are combined in a simple model. This model shows the interaction of an instrument attribute and an SSR attribute. The instrument transitions between `Pointing, Idle, Calibrating` and `Take-Image`. The SSR transitions between `Recording, Playback` and `Idle`. The time required for `Pointing, Calibrating, Recording` and `Playback` activities are constrained by the parameters of those activities. In addition, `Take-Image` and `Recording` activities must be simultaneous, and whenever a `Playback` occurs on the SSR the instrument must be `Idle`.

A complete EUROPA plan consists of a a set of *timelines*, one for each attribute, each comprised of a sequence of tokens. Figure 4 illustrates a small EUROPA plan involving two satellites, and a TDRSS communication satellite for downlinking data. The figure indicates that every `Take-Image` activity is synchronous with a `Record(`N`)` activity on the associated SSR, where N is a parameter standing for the amount of data added to storage. Similarly, every `Playback(`N`)` activity for a satellite is synchronous with a `Contact` activity when TDRSS is in contact with that satellite. Activities such as `Aiming` the antenna are also shown.

The EUROPA planner supports object-oriented descriptions of models. Most subsystems of satellites are quite similar, so we expect that we can define a relatively large number of different satellites quite easily. We can then vary the parameters of these different satellite models to create more or less challenging EOS domains. For instance, we can vary the transmission rates and SSR capacities of the satellites, the number of ground stations or TDRSS contacts, as well as change the instrument makeup of satellites, to assess the impact of different scenarios for particular sets of requests.

V. The Search Algorithm

In theory, the optimal solution to an observation scheduling problem can be found using the well known systematic Branch and Bound algorithm. Unfortunately, complete search algorithms are simply not practical for most large scheduling problems. Bensana et al. [2] indicate that they were unable to optimally solve problems with more than about 200 observations using Russian Doll Search (a clever but specialized variation on Branch and Bound). Pemberton [12] makes similar observations. The only alternatives are to use some form of greedy search or hill-climbing search, possibly augmented with stochastic variation to escape local optima. Fortunately, for observation scheduling these approaches tend to work well, because there are usually many local optima that are nearly as good as the global optimum. Thus, by injecting stochastic variation into a greedy search procedure one of these reasonably good solutions can often be found very quickly.

For our purposes, we have chosen to overlay a stochastic greedy search algorithm on the constraint-based planning techniques discussed earlier. In particular, the greedy search will choose and schedule observations, and the constraint based planning foundation will 1) propagate constraints to rule out possibilities inconsistent with each observation assignment, and 2) expand individual observations by including any necessary setup and postprocessing steps required by the scheduled observations. The stochastic greedy search algorithm is based on the HBSS algorithm developed by Bresina [3]. The basic algorithm for HBSS looks like a simple greedy search with restarts.

A modified version of the algorithm appears in Figure 5. This algorithm repeatedly selects an observation that still has time windows available, then selects a time to schedule the observation. This assignment is added to the plan,

```
HBSS
  repeat
    while observations are still possible
      Select an observation
      Select a time for the observation
      Assign the observation to the time slot
      Propagate constraints
    end while
    Expand any remaining subgoals
    Check for consistency
  end repeat
end
```

Figure 5: A sketch of the HBSS algorithm modified for the EOS Scheduling problem.

and constraint propagation takes place to infer simple consequences of the newly scheduled observation. These inferences include eliminating choices for observations and otherwise eliminating the values of variables in the DCSP representation of the plan. In addition, if the scheduled observation requires setup steps (or has other preconditions that must be established) this expansion can (optionally) be done at this time. The resulting constraint propagation may lead to an inconsistency, meaning that the scheduling of this particular observation in this time slot is not possible and must be undone.

When it is not possible to schedule any more observations, the resulting schedule must be examined to make sure that all observations having subgoals (setup steps or other preconditions) have been completely expanded. If not, this expansion must be done, constraints propagated, and the resulting schedule again checked for consistency. If this process is successful, the resulting schedule is returned. The algorithm then restarts, and builds another schedule.

What distinguishes the HBSS algorithm from ordinary greedy search is the way in which observations and time slots are selected. In a pure greedy search, these choices are made absolutely by a heuristic. In the HBSS algorithm, the heuristic must rank or score the possible alternatives. HBSS then chooses probabilistically from among the alternatives, weighted according to their ranking or score. Thus, possibilities ranked highly by the heuristic have higher probability of being selected, but other lower ranked possibilities are sometimes selected. This means that several alternatives with roughly the same score will have roughly equal probability of being chosen. Because of the stochastic character of the selection steps, alternative schedules are likely to be explored with each successive restart of the algorithm.

Like most search procedures, the effectiveness of HBSS depends critically on the quality of the heuristic advice. Bresina [3] has shown that HBSS is particularly effective when the ranking heuristics typically give good advice. As the quality of the heuristic advice declines, HBSS must search progressively longer (more restarts) to find near optimal schedules. In the next section we develop contention heuristics for ranking observation choices.

VI. Contention Heuristics

The success of greedy search methods depends largely on the heuristic used to decide which variable to assign next, and which value to assign to that variable. For observation scheduling, these steps correspond to selecting the observation to schedule next, and selecting the time slot for the observation. An obvious heuristic for choosing an observation is to select the one with the highest priority. In general, this will ensure that the schedule is loaded with as many high priority observations as possible before any lower priority observations are considered. However, there may be many observations with the same priority, and the order in which we consider these observations can have a dramatic impact on the resulting schedule. Consider the simple example shown in Figure 6, where there are two observations, A and B, of equal priority. As shown, there are several opportunities for scheduling A, but only one opportunity for scheduling B, which overlaps with the first opportunity for A. If we choose observation A first, and foolishly schedule it in the first available time slot, then observation B will not appear in the schedule. In contrast, if we were to schedule B first, other opportunities would still remain for observation A.

Figure 6: The impact of variable and value ordering. `Take-Image A` has three possible timeslots, while `Take-Image B` has only 1. The temporal constraints imply that scheduling `Take-Image A` at time 1 makes it impossible to schedule `Take-Image B` at all, since it can only start at time 2.

This example suggests a simple rule of thumb for choosing which observation to schedule next: *prefer observations having the fewest remaining opportunities.* This heuristic resembles the Minimum Remaining Values (MRV) heuristic commonly used in the CSP community [7]. Calculating the number of remaining opportunities for an observation is appealing because it is simple to compute, and provides at least some estimate of how easy it is to schedule that particular observation. However, it does not give any estimate of how much "contention" there is for those opportunities. For example, if there are two remaining opportunities for a high priority observation, but absolutely no contention for one of the time slots, then the observation will always be easy to schedule. In contrast, if there are numerous other observations that could use those time slots, then there is good reason to schedule the observation early, to make sure it gets one of those time slots.

This leads us to a more sophisticated measure of contention. To start with, we will only consider contention for time slots. We first define some terms: Observations(t) is the set of observations that could occur at time t, and Opportunities(o) is the set of discrete opportunities for observation o (noting that each discrete opportunity is exactly long enough to accommodate the observation.) For a given time slot, we could measure contention by counting the number of observations that want that time slot, weighted by the priority of the observation:

$$\text{Contention}(t) = \sum_{o \in \text{Observations}(t)} \text{Priority}(o)$$

However, this measure doesn't incorporate how badly each observation needs the time slot; i.e. if an observation can be scheduled in only that time slot, it needs the time slot badly, but if it can be scheduled in lots of different time slots, it doesn't need the time slot very badly at all.

We can define the *need* of an observation as:

$$\text{Need}(o) = \frac{\text{Priority}(o)}{|\text{Opportunities}(o)|}$$

The *contention* for a particular time slot can then be defined as:

$$\text{Contention}(t) = \sum_{o \in \text{Observations}(t)} \text{Need}(o)$$

The *contention* for a particular observation can then be defined as:

$$\text{Contention}(o) = \min_{t \in \text{Opportunities}(o)} \text{Contention}(t)$$

We take the minimum here because if there is a low-contention opportunity to schedule an observation, this should not be overshadowed by other higher contention opportunity. In other words, adding another opportunity for an observation should never increase the contention measure for that observation.

In developing the equations above, we regarded observations as if they only required a single scene or time slot. For observations that involve a sequence or group of scenes we would have to sum up (or maximize over) the contention measures for each of the individual scenes (time slots). With pointable instruments, there is an interval during which a given scene could be taken. This can also be incorporated by generalizing the above equations to use time intervals rather than time points. The resulting contention measures have some similarity to measures of *slack* used in job shop scheduling problems [17, 1].

Measuring contention for a global resource like SSR capacity involves generalizing the above contention measure to consider the amount of the resource needed by an observation, the resource capacity, and the interval of time under consideration. Let Requires(o, r) be the amount of resource r required by observation o, and let Capacity(r, i) be the capacity of a resource r over a time interval i. Thus, an SSR with a capacity of 50 has a Capacity$(r, i) = 50$. If a playback of 20 units occurs within the interval i, then Capacity$(r, i) = 70$. We then generalize the above definitions to be:

$$\text{Need}(o, r) = \text{Requires}(o, r)\frac{\text{Priority}(o)}{|\text{Opportunities}(o)|}$$

$$\text{Contention}(r, i) = \frac{\sum_{o \in \text{Observations}(i)} \text{Need}(o)}{\text{Capacity}(r, i)}$$

$$\text{Contention}(r, o) = \min_{i \in \text{Opportunities}(o)} \text{Contention}(r, i)$$

Again, note that these measures change as activities are scheduled. In particular, as activities that empty the SSR are scheduled, Capacity(r, i) may increase, and as observations are scheduled, Capacity(r, i) may decrease.

Intuitively, these contention measures provide a more accurate assessment of how hard it is to actually schedule an observation. Using these measures, our variable ordering heuristic is:

> *Schedule the observation of highest priority and highest overall contention*

where contention will be a weighted sum of contention measures for the different resources, such as time slots and SSR capacity. This approach assumes that resources are *independent*; while not true, it does provide an efficiently computable approximation. This heuristic provides a ranking of observations suitable for use with the HBSS search procedure.

Given an observation to schedule, we would prefer to put it in the place where it will compete with the fewest other observations. We can use the above contention measures to define a value ordering heuristic:

> *Schedule an observation in the opportunity with the least contention*

Again, this heuristic provides a ranking suitable for use with the HBSS search procedure.

VII. Conclusions and Future Work

We have presented the problem of scheduling observations on a collection of Earth Observing Satellites and discussed a candidate representation and solution methodology. In order to produce good plans, we advocated a high-fidelity model incorporating both satellite resources and communications resources. In order to gain maximum flexibility in solving problems, we used the CBI paradigm, which allows us to use algorithms from the DCSP community. We believe that this problem is large enough and complex enough that a biased greedy stochastic search method is the best approach. We have presented a heuristic for guiding this search procedure based on a general notion of contention for resources.

Our next tasks are to finish the implementation of this heuristic, select the bias function to be used with HBSS, and select the exact method by which subgoals of scheduled observations will be inserted into the plan. Once this is done, we can then begin experiments to test the effectiveness of this procedure on large, heterogeneous EOS scheduling problems.

References

[1] J. C. Beck and M. Fox. A generic framework for constraint-directed search and scheduling. *AI Magazine*, 19(4):101–130, 1998.

[2] E. Bensana, M. Lemâitre, and G. Verfaillie. Earth observation satellite management. *Constraints*, 3(4), 1999.

[3] John Bresina. Heuristic-biased stochastic search. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[4] S. Burrowbridge. *Optimal Allocation of Satellite Network Resources*. PhD thesis, Virginia Tech, Virginia, 1999.

[5] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–94, 1991.

[6] J. Frank, A. Jónsson, and P. Morris. On the representation of mutual exclusion constraints for planning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation*, 2000.

[7] R. Haralick and G. Elliot. Increasing tree search efficiency for constraint satisfaction. *Journal of Artificial Intelligence*, 14:263–313, 1980.

[8] A. Jónsson and J. Frank. A framework for dynamic constraint reasoning using procedural constraints. *Proceedings of the Euopean Conference on Artificial Intelligence*, 2000.

[9] Ari K. Jónsson, Paul H. Morris, Nicola Muscettola, Kanna Rajan, and Ben Smith. Planning in interplanetary space: Theory and practice. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[10] M Lemâitre, G. Verfaillie, F. Jouhaud, J. Lachiver, and N. Bataille. How to manage the new generation of agile earth observation satellites? In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2000.

[11] H. Muraoka, R. Cohen, T. Ohno, and N. Doi. Aster observation scheduling algorithm. In *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems*, 1998.

[12] J. Pemberton. Towards scheduling over-constrained remote sensing satellites. In *Proceedings of the 2d International Workshop on Planning and Scheduling for Space*, 2000.

[13] W. Potter and J. Gasch. A photo album of earth: Scheduling landsat 7 mission daily activities. In *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems*, 1998.

[14] G. Rabideau, R. Knight, S. Chien, A. Fukanaga, and A. Govindjee. Iterative planning for spacecraft operations using the ASPEN system. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 1999.

[15] R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, and A. Fukanaga and. Using ASPEN to automate EO-1 activity planning. In *Proceedings of the IEEE Aerospace Conference*, 1998.

[16] D. Smith, J. Frank, and A. Jónsson. Bridging the gap between planning and scheduling. *Knowledge Engineering Review*, 15(1), 2000.

[17] S. Smith and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 139–44, 1993.

[18] W. Wolfe and S. Sorensen. Three scheduling algorithms applied to the earth observing domain. *Management Science*, 46(1), 2000.